REMARKS

Claims 1-30 were presented for examination. Claims 1-30 have been rejected under 35 U.S.C. § 103(a). Claims 1, 2, 4, 5, 10, 11, 13, 14, 19 and 23 have been amended. Claims 31-35 have been added. No new matter has been added. The following comments address all stated grounds for rejection, and place the presently pending claims 1-35 in condition for allowance.

I. Claims Amendments

Independent Claims 1, 10, 19 and 23 have been amended to clarify that the hypertext link is provided in a listing of the source code to associate an element of the source code with an element of the block diagram model.

Dependent Claims 2, 4, 5 have been amended to be consistent with the independent Claim 1. Dependent Claim 11 has been amended to be consistent with the independent Claim 10. Dependent Claim 14 has been amended to be consistent with the independent Claim 13.

New Claims 31-35 have been added to provide additional features of the present invention. More specifically, Claims 31-34 provide that the block diagram model is an executable block diagram model and Claim 35 provides that the step of providing a hypertext link in the listing of the source code, as required in Claim 1, is provided by replacing an element of the source code with a hypertext link.

II. Claim Rejections Under 35 U.S.C. § 103(a)

A. Claims 1-5, 9-14 and 18-29 Rejected under 35 U.S.C. § 103(a) as Unpatentable

Claims 1-5, 9-14, 18-29 are rejected under 35 U.S.C. § 103(a) as unpatentable over U.S. Patent No. 6,269,475 to Farrell et al. ("Farrell") in view of U.S. Patent No. 6,792,418 to Binnig et al. ("Binnig"). Claims 1, 10, 19, 23 and 27 are independent claims. Applicants respectfully traverse these rejections.

1. Independent Claims 1, 10, 19 and 23 Rejected under 35 U.S.C. § 103(a)

Claims 1, 10, 19 and 23 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Farrell in view of Binnig. Applicants respectfully traverse these rejections and contend that

neither Farrell nor Binnig, alone or in combination, render claims 1, 10, 19 and 23 unpatentable.

Independent claims 1, 10, 19 and 23 are directed to generating source code corresponding to a block diagram model. The code generation converts the source model language into a programming language, such as C or C++. This allows the block diagram model to be represented in a programming language that can be executed on a target processor, thereby implementing the functionality of the specified block diagram model.

The independent claims 1, 10, 19 and 23 further enable providing a hypertext link in a listing of the generated source code that associates an element of the generated source code with an element of the block diagram model. A hyperlink is a selectable connection that when selected results in electronic navigation to the element of the block diagram model to which the source code is related.

Farrell teaches an object oriented program editor ('Editor"). The Editor discussed in Farrell creates an object model to represent the source code and codeblocks. Object models represent recognized elements in a programming language, such as a class. Codeblocks can be represented as object models and object models can be represented as codeblocks. The object models and codeblocks are merely ways of representing the same programming language, whether graphically or textually. The object models simply aid the user in developing source code and are not used to generate a programming language from a block diagram model language. The Editor allows the user to edit the source code and codeblocks using the text editor or the graphical display. A codeblock is defined by Farrell as a sequence of text characters representing an instance of a syntactic element of a programming language, where a syntactic element is defined as a unit of source code. In other words, the textual language creates a tree of syntactic elements, and, therefore, a tree of codeblocks. Codeblocks can contain pointers to the source code as well as to the object model. The pointers discussed in Farrell are used to indicate the position of the codeblock in the source code and object model.

Binnig teaches a database manager and system using a fractal hierarchical index structure. It is directed to a scheme for adding files to a database; a scheme for acting upon files in a database and; a scheme for effective searching and discovery of files in a database. It defines

"access pointer" to refer to a relative or absolute pointer that points to a file in a file structure or file directory, or a sequence number or coordinates for the retrieval of a file from a database. The access pointer provides a physical or logical link between a semantical unit representing the file and the file, and may describe the physical or logical location where the file is stored in the database. That is, an access pointer refers to the memory address of the file in the database. When the user is looking for a file they will input a string. When the input string is deemed to be associated to a particular semantical unit the user can act upon the corresponding database file by using the access pointer that is associated to the semantical unit. Binnig suggests that access pointers can be hypertext links.

Both Farrell and Binnig fail to teach or suggest providing a hypertext link in a listing of the generated source code to associate an element of the generated source code with an element of the block diagram model. In the present illustrative teachings, a hypertext link provides a selectable connection that associates an element of the generated source code with an element of the block diagram model for navigation from the source code to the block diagram model. The source code is first generated and then a hypertext link is provided in the listing of the source code. Farrell teaches pointers in a program that allows pointing to memory locations in the code for the development of the code, such that changes to the data in the memory locations that are pointed to by the pointer are mirrored to the pointer. Binnig teaches hypertext links for file retrieval in a database. The hypertext link, of the present claims of issue, is not a pointer in a programming language that allows manipulation of the data stored in the memory location pointed to by the pointer. Rather, the hypertext link, of the present claims of issue, provides a reference in a list of the generated source code that links text in the listing of the source code with a specific element in the block diagram model to allow navigation from the source code to the block diagram model. As such, Farrell and Binnig fail to teach or suggest providing a hypertext link in generated source code to associate an element of the source code with an element of the block diagram model.

For at least the aforementioned reasons, neither Farrell nor Binnig alone or in combination teach or suggest generating source code corresponding to a block diagram model, and providing a hypertext link associating an element of the generated source code with elements of the block diagram model. Claims 2-5 and 9 depend on Claim 1. Claims 11-14 and 18 depend on claim

10. Claims 20-22 depend on claim 19. Claim 24-26 depend on claim 23. Thus, neither Farrell nor Binnig alone or combined detract from the patentability of claims 1-5, 9-14 and 18-26. Accordingly, Applicants respectfully request the Examiner to reconsider and withdraw the Examiner's rejection of claims 1-5, 9-14 and 18-26 under 35 U.S.C. § 103(a).

2. Independent Claim 27 Rejected under 35 U.S.C. § 103(a)

Claim 27 stands rejected under 35 U.S.C. § 103(a) as unpatentable over Farrell in view of Binnig. The Applicants respectfully traverse this rejection.

Independent claim 27 recites a method for providing source code identifying an element of a graphical model, and generating a document with information about the source code. This claim further recites providing, in a document, a hyperlink referencing the element of the graphical model. That is, the method generates a document about source code identifying an element of the graphical model and the document includes a hyperlink referencing the element of the graphical model. The document generated provides a mark-up language report. The mark-up language report can contain, for example, information about the graphical model, settings of the code generator and the generated program source code in syntax highlighted form.

Neither Farrell nor Binnig teach or suggest providing source code identifying an element of a graphical model and generating a document with information about the source code. In contrast to the present claim of issue, Farrell discusses that source code can be compiled to produce an executable code file. Such an executable code file does not provide a report about the graphical model, it simply is a file that allows the code to be executed. The present claim of issue, however, provides a document that can contain information about the graphical model, settings of the code generator and the generated program source code in syntax highlighted form. As such, the document generated by the present claim of issue is distinguished from the executable code file generated by the compiler of Farrell. Binnig does not teach generating a report from source code, but rather, teaches a searchable database.

Further, both Farrell and Binnig fail to teach or suggest providing, in the document, a hyperlink referencing the element of the graphical. In the present illustrative teachings, a hypertext link provides a selectable connection that associates an element of the generated

source code with an element of the graphical model for navigation from the source code to the graphical model. The document is first generated and then a hypertext link is provided in the document. Farrell teaches pointers in a program that allows pointing to memory locations in the code for the development of the code, such that changes to the data in the memory locations that are pointed to by the pointer are mirrored to the pointer. Binnig teaches hypertext links for file retrieval in a database. The hypertext link of the present claim is not a pointer in a programming language that allows manipulation of the data stored in the memory location pointed to by the pointer. Rather, the hypertext link, of the present claim of issue, provides a reference in a list of the generated source code that links text in the listing of the source code with an element in the graphical model to allow navigation from the source code to the graphical model. As such, Farrell and Binnig fail to teach or suggest providing, in the document, a hyperlink referencing the element of the graphical model.

In addition, Applicants maintain there is no motivation or suggestion to combine Farrell and Binnig. As discussed above Binnig teaches a searchable database and Farrell an object oriented program editor. The object oriented program editor of Farrell provides software developers with an environment for developing source code. The searchable database of Binnig allows users of the database to search for elements efficiently. As such, Farrell addresses issues faced in software development and Binnig addresses issues faced in searching databases. It would not be obvious to one skilled in the art, at the time the invention was made to when developing a programming editor to incorporate a searchable database.

For at least the aforementioned reasons, neither Farrell nor Binnig teach or suggest providing source code identifying an element of a graphical model and generating a document, comprising information about the source code. Nor do Farrell or Binnig teach or suggest providing, in a document, a hyperlink referencing the element of the graphical model. Claims 28-29 depend on claim 27. Thus, neither Farrell nor Binnig alone or combined detract from the patentability of claims 27-29. Accordingly, Applicants respectfully request the Examiner to reconsider and withdraw the Examiner's rejection of claims 27-29 under 35 U.S.C. § 103(a).

B. Claims 6-8, 15-17 and 30 Rejected under 35 U.S.C. § 103(a) as Unpatentable

Claims 6-8, 15-17 and 30 are rejected under 35 U.S.C. § 103(a) as unpatentable over

Farrell in view of Binnig in further view of U.S. Publication No. 2002/0055891 to Yang ("Yang"). The Applicants respectfully traverse this rejection.

1. Dependent Claims 6-8, 15-17 and 30 Rejected under 35 U.S.C. § 103(a) as Unpatentable

Dependent claims 6-8, 15-17 and 30 stand rejected under 35 U.S.C. § 103(a) as unpatentable over Farrell in view of Binnig in further view of Yang. Claims 6-8, 15-17 and 30 depend on claims 1, 10 and 27, respectively and incorporate all the patentable limitations of these independent claims.

As discussed above, neither Farrell nor Binnig detract from the patentability of independent claims 1, 10, 19, 23 and 27. Yang does not detract from the patentability of independent claims 1, 10, 19, 23 and 27. Yang discusses an electronic catalogue utilizing 3D image display to provide a researching method and researching system for interests in commercial goods. Yang discusses that the electronic catalogue documents may be prepared in Markup languages such as HTML, XML and SMGL. Yang fails to teach or suggest generating source code corresponding to a block diagram model and generating hypertext links associating source code with a block diagram model. As such, Yang fails to bridge the factual deficiencies of Farrell and Binnig.

For at least the aforementioned reasons, neither Farrell, nor Binnig, nor Yang, alone or in combination, teach or suggest all the elements of claims 1, 10, 19, 23 and 27. Claims 5-7 depend on claim 1. Claims 15-17 depend on claim 10. Claim 30 depends on claim 27. Thus, neither Farrell nor Binnig nor Yang alone or combined detract from the patentability of claims 5-7, 15-17 and 30. Accordingly, Applicants respectfully request the Examiner to reconsider and withdraw the Examiner's rejection of claims 1-30 under 35 U.S.C. § 103(a).

III. New Claims

Applicants add new Claims 31-35. New independent Claim 31 provides for generating source code from an executable block diagram model. Claim 32-34 depend on Claim 31, and therefore incorporate all of the patentable features of Claim 31. New Claim 35 depends on

Claim 1, and therefore incorporates all of the patentable features of Claim 1. Applicants contend that the new Claims 31-35 are allowable over the cited references.

As to Claims 31-34, both Farrell and Binnig fail to teach or suggest generating source code corresponding to an executable block diagram model. Farrell, however, discusses developing and representing a textual programming language both graphically and textually, but does not teach or discuss an executable block diagram. Rather, Farrell discusses object models to represent textual language. Binnig simply discusses a searchable database, and does not disclose generating code from a block diagram model.

As to Claim 35, both Farrell and Binnig fail to teach or suggest replacing an element in the source code listing with the hypertext link to associate an element of the source code with an element of the block diagram model. Rather, Farrell discusses pointers in a program to passing data and Binnig discusses a database that uses hypertext links to retrieve files. As such neither Farrell nor Binnig teach or suggest replacing an element in the source code listing with a hypertext link.

For at least the aforementioned reasons, none of the cited references detract from the patentability of Claims 31-35. Accordingly, Applicants respectfully request the Examiner to place new Claims 31-35 in condition for allowance.

IV. Conclusion

In view of the above amendment, applicant believes the pending application is in condition for allowance.

Dated: October 26, 2006

Respectfully submitted,

By David R Burns

David R. Burns
Registration No.: 46,590
LAHIVE & COCKFIELD, LLP
28 State Street
Boston, Massachusetts 02109
(617) 227-7400
(617) 742-4214 (Fax)
Attorney/Agent For Applicant